



Señales y Sistemas

PRÁCTICA 3: Introducción al Filtrado Digital

Profesora responsable de esta práctica: Borja Imaz Lueje

Bibliografía:

Signals and Systems; Alan V. Oppenheim, Alan S. Willsky, 2ª edición (Agosto 1996)
Prentice Hall; Capítulo 1.

A. V. Oppenheim, R. W. Schaffer, “Discrete-Time Signal Processing”, 2ª Ed., Upper Saddle River, NJ: Prentice-Hall, 1999.

Procesado digital de la señal para graduados en Telecomunicaciones; ULPGC, M. A. Ferrer Ballester, A. Morales Moreno y J.B. Alonso-Hernández, 2013.

Aprenda MatLab como si estuviera en primero; Javier García de Jalón; Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid.

Metodología de trabajo:

- Antes de acudir a clase, resulta necesario haber trabajado la práctica, la sesión de prácticas se utilizará para resolver dudas. No es posible realizar todos los ejercicios únicamente durante la clase de prácticas.
- No es necesario entregar ningún guion de prácticas al finalizar la clase.
- Recuerde que si en la ventana de Matlab teclea `doc nombre_comando` o `doc nombre_comando` el programa le ofrecerá ayuda sobre la el significado y la forma de utilizar el comando `nombre_comando`.
- En la mayoría de los casos se le proporciona el código completo y su labor será entenderlo. No obstante, en algunos casos usted deberá escribir parte del código.

El fichero *voz.wav* contiene la señal de voz que vamos a filtrar. Para obtener dicha señal ejecute el siguiente comando en Matlab:

```
[mySpeech, fs]=audioread('voz.wav');
```

¿Qué contiene *mySpeech*? ¿Qué contiene *fs*?

Para escuchar dicha señal de voz, ejecute:

```
soundsc (mySpeech) ;
```

¿Qué ocurre?

Represente dicha señal en matlab (`plot`, `stem`).

A partir de la representación, encuentre el principio y final de una zona de la señal sin presencia de voz y de otra zona con voz. Seleccione 512 muestras de cada una de ellas y guárdelas en dos señales (*trozo_sinVoz* y *trozo_conVoz*).

Represente en dBs el módulo de la TF (vía DFT) del tramo seleccionado sin voz en el intervalo $[0, \pi)$. Normalice el eje de frecuencia por π . Debería obtener los valores de $20 \cdot \log_{10}(\text{abs}(\text{trozo_sinVoz}))$ en el intervalo $[0, 1)$. Tal y como se indica en las transparencias del tema 6 (DFT), la DFT en Matlab se define de la forma:

$$X_N = \text{fft}(x, N)$$

Donde x es un vector con la secuencia de valores y N el número de puntos a utilizar en la DFT. Todas las TFs (vía DFTs) que tenga que realizar a lo largo de esta práctica se harán en dBs y con la normalización indicada.

Observe que aparece un pico por encima del resto de los valores. Dicho pico corresponde a un tono que se superpone a la señal de voz. Teniendo en cuenta el valor de f_s , estime la frecuencia del tono superpuesto a la señal en Hz.

El objetivo de la práctica es eliminar la presencia del tono en la señal de voz. Para eliminar el tono interferente de la voz se utilizará un filtro de tipo banda eliminada centrado en la frecuencia del tono interferente. Esto nos lleva a diseñar diferentes filtros de tipo banda eliminada (p.e., Chebyshev, Kaiser) y quedarnos con el que obtenga una señal de voz más “limpia”.

Del tramo seleccionado de voz con tono interferente (*trozo_conVoz*), calcule la magnitud de su TF de la misma manera que se indicó para el trozo de señal sin voz. Con ayuda de dicha gráfica defina los valores de las frecuencias que delimitan las bandas de transición del filtro ($Wp1$, $Ws1$, $Ws2$, $Wp2$). Considere que el rizado en la banda de paso es $\delta_1 = 0.3$, y el rizado en la banda atenuada es $\delta_2 = 0.0009$. Eso significa que las atenuaciones asociadas a dichas bandas vendrán dadas en dBs por:

```
Rp=20*log10(1-delta1);
```

```
Rs=20*log10(delta2);
```

Defina:

```
W_i =[0 Wp1 Ws1 Ws2 Wp2 1];
```

```
H_i =[Rp Rp Rs Rs Rp Rp];
```

Utilizando `plot(W_i, H_i, 'g')` dibuje las especificaciones deseadas, es decir, el filtro ideal considerado. Represente en una sola gráfica el espectro de la voz con tono interferente y las especificaciones del filtro normalizadas para que la amplitud máxima sea la unidad. Es decir, represente en la misma gráfica el espectro de *trozo_conVoz* y el filtro ideal considerado. Nota: puede utilizar los comandos `hold on` y `hold off`.

Recuerde que el objetivo de esta práctica es aprender las características que diferencian un filtro de otro; así, no utilice especificaciones demasiado estrictas como lo son rizados muy pequeños o anchos de banda muy estrechos.

Diseñar filtros IIR para eliminar el tono interferente.

Vamos a diseñar un filtro IIR que cumplan las especificaciones (W_i , H_i) anteriores mediante el método de la transformación bilineal. Considere un filtro de Chebyshev tipo II. Ejecute:

```
Wp= [Wp1 Wp2];  
Ws= [Ws1 Ws2];  
  
[n, Wn] = cheb2ord(Wp, Ws, Rp, -Rs);
```

¿Qué nos da n ?

Calcule los coeficientes de dicho filtro mediante la función *cheby2* de Matlab. Para usarla correctamente puede consultar la ayuda de matlab (`help cheby2`).

Represente la respuesta al impulso del filtro diseñado. Represente la magnitud (en dBs) de la respuesta en frecuencia del filtro junto con las especificaciones deseadas. Puede utilizar la función *freqz* de matlab.

Represente también el diagrama de polos y ceros. Puede utilizar la función *zplane* de matlab. ¿Cumple el filtro las especificaciones?

Utilizando la función *filter*, calcule la salida del filtro de Chebyshev tipo II diseñado cuando la entrada es la señal *mySpeech*. Reproduzca el sonido de la señal obtenida tras filtrar. ¿Se escucha el tono?. Calcule y dibuje la magnitud (en dBs) de la TF (vía DFT) de un tramo con voz. ¿Se ha eliminado el tono interferente? Justifique su respuesta.

Diseñar filtros FIR para eliminar el tono interferente.

En este apartado se trata de diseñar un filtro FIR que aproxime las especificaciones (W_i , H_i) mediante el método del enventanado. Recuerde que para utilizar este método, primero hay que elegir la ventana a utilizar, calcular sus parámetros para que el filtro cumpla las especificaciones indicadas y, finalmente, enventanar la respuesta al impulso ideal.

Suponiendo que se utiliza una ventana de Kaiser, estime los parámetros necesarios (Ω_c , longitud M de la ventana, y la β) para que el filtro cumpla las especificaciones deseadas. Para ello vamos a utilizar las funciones *kaiserord* y *kaiser* de Matlab, de la siguiente manera:

```
bandas = [Wp1 Ws1 Ws2 Wp2]*(fs/2);  
delta1=0.05;  
delta2 = 0.0009;  
[M,wn,beta,tipo]= kaiserord([bandas],[1 0 1],[delta1 delta2  
delta1],fs);
```

Con la función *fir1* de Matlab calculamos los coeficientes del filtro:

```
M = M + rem(M,2);  
h=fir1(M,wn,tipo,kaiser(M+1,beta),'noscale');
```

Represente la respuesta al impulso del filtro diseñado. Calcule la magnitud de la respuesta en frecuencia (función *freqz* de matlab) del filtro y represéntela en dBs. Represente también su diagrama de polos y ceros.

Utilizando la función *filter*, calcule la salida del filtro de Kaiser diseñado cuando la entrada es la señal *mySpeech*. Reproduzca el sonido de la señal obtenida tras filtrar. ¿Se escucha el tono?. Compare con el resultado obtenido por el filtro IIR diseñado anteriormente. Calcule y dibuje la magnitud (en dBs) de la TF (vía DFT) de un tramo con voz. Escuche la salida del filtro. ¿Se ha eliminado el tono interferente? Justifique su respuesta. Compare los resultados obtenidos con ambos tipos de filtros.

(OPCIONAL) Diseñar filtros IIR y FIR con *filterDesigner*

El diseño de ambos tipos de filtros se puede realizar con la herramienta *filterDesigner* de matlab. Los pasos a realizar son los siguientes:

1. Diseñar el filtro con *filterDesigner*.
2. Exportarlo al espacio de trabajo como Objeto.
3. Filtrar la señal de voz: `VozFiltrada = filter(Hd, mySpeech)`

Realice el diseño de los dos tipos de filtros considerados en los apartados anteriores con *filterDesigner* y calcule la señal a la salida de cada filtro siguiendo los pasos anteriores. ¿Obtiene los mismos resultados?

Pruebe a diseñar y filtrar la señal de voz propuesta con otros filtros IIR (butterworth, chebyshev tipo I, elíptico) y otros filtros FIR (basados en otras ventanas). Compare y comente los resultados obtenidos.